COW, a Flexible Platform for the Enactment of Learning Scenarios

Thomas Vantroys, Yvan Peter

COW, a Flexible Platform for the Enactment of Learning Scenarios

Thomas Vantroys[1,2] and Yvan Peter[1]

[1] TRIGONE Laboratory, NOCE Team
Bât. B6 – Cité Scientifique,
59655 Villeneuve d'Ascq, France
{Thomas.Vantroys, Yvan.Peter}@univ-lille1.fr
http://noce.univ-lille1.fr

[2] Archimed SA,
49 Boulevard de Strasbourg,
59042 Lille, France
t.vantroys@archimed.fr
http://www.archimed.fr

Abstract. Open and Distance Learning platforms are more than system delivering pedagogical ressources. They require mechanisms for the enactment and coordination of pedagogical modules and learning activities. A common solution to express learning paths in learning management systems (LMS) can be the use of Educational Modelling Languages (EML). The next step will be the enactment of these models. For that purpose, workflow management system can be used. These systems formerly reserved for highly structured procedures can be used in dynamic and reactive environments such as virtual campuses platforms, thanks to a enhanced flexibility in the execution of models and in the management of exceptions. In this article we shall present COW our flexible workflow engine dedicated to open and distant learning. We will compare EML and workflow approach and see how to pass from a pedagogical modelisation to a workflow modelisation. We shall see how it is possible to organize the pedagogical modules and the learning paths to answer the expectation within the framework of individual courses (lifelong learning orientation) or within the framework of group courses (closer to the traditional face to face learning).

1 Introduction

The model driven approach is gaining momentum. We can see that with the Model Driven Architecture (MDA) [1] promoted by the Object Management Group [2], which aims to separate the model of a system from its implementation. The main reason is that a change in the technology field, like passing from Java to C#, should not have consequences on the model because models are persistent and implementation are transient. The same approach can be conducted in learning management systems (LMS). The model of a course is independant

from its execution platform. To express models, Educational Modelling Languages have emerged like PALO [3] or EML [4], and a work on standardization has began in the CEN/ISSS (Information Society Standardization System) [6]. The next step is the execution of these models. A possible approach can be the use of flexible workflow systems [10], [11]. Indeed, the pedagogical sciences teach us to structure and coordinate the learning activities and as explained in [11] learning is process-oriented.

Managing the schedule of the activities performed by students can be a daunting task for tutors and teachers especially if there are a large number of students. They have to keep track of each student's progress to provide them with new activities. Workflow systems are dedicated to this kind of work. However, even in case of virtual classroom style of distance learning, one would like to be able to adapt the learning path of a particular student to better fit his needs. This kind of flexibility is unfortunately not well handled by current workflow systems. For this reason, we have decided to start the development of a flexible workflow system dedicated to distance learning. In this article, we will present The work we have done to build a flexible workflow system suited to LMSs and our reflexions about the enactement of Educational Modelling Languages (or more precisely IMS-LD) by our workflow system. We will describe how pedagogical modules and learning paths can be organized to support both individual learning paths (lifelong learning orientation) and group based courses (closer to the traditional face to face learning).

The next section will present more precisely what are the purpose of the EMLs. We will notably see the different category of languages which compose the EMLs. We will emphasize on IMS Learning Design (IMS-LD) which is, from our perspective, one of the best approach to describe a learning scenario. Then in section 4 we will present workflow systems in general and more precisely the modelisation aspect in the standard of the Workflow Management Coalition (WfMC). After that, we will present COW, our flexible workflows system dedicated to the enactment of learning scenario. We will explain the base requirements and how we realize them.Section 7 will show the existence of a common concepts between learning flow and work flow and so the links we can weave in order to transform an IMS-LD scenario into a workflow process. Then we will present how the course models are instantiated and enacted. Section 9 will conclude our paper by summarizing our contribution and by reviewing the perspectives.

2 Educational Modelling Languages

2.1 Presentation

Teaching is not just delivering information. Current platforms focus more on delivering pedagogical resources than on the pedagocial aspect of learning. Standardization efforts are now moving from content delivery ressources to Educationnal Modelling Languages (EML).

We take as definition for an EML the one provided by [6]:

"An EML is a semantic information model and binding, describing the content and process within a 'unit of learning' from a pedagogical perspective in order to support reuse and interoperability"

EMLs groups together many languages with differents aims. There are languages designed to express questionnaire and assesment like the *Tutorial Markup Language* (TML) [5] or the *IMS Question & Test Interoperability* (IMS-QTI) [8].

We will explain more in details the IMS Learning Design Specification because it is for us the language offering the most easy way to express a learning path in a pedagocial neutral way. Futhermore, it seems on a good way to become a de facto standard.

2.2 IMS Learning Design Specification

The IMS Learning Design Specification (IMS-LD) [7] was published at the end of january 2003. It is based on the work done by the Open University of Netherlands on Educational Modelling Language [4]. This language expresses the "learning flow", i.e., how the different learning activities are linked and the resources they use. A Learning design describes how learning objectives can be reached.

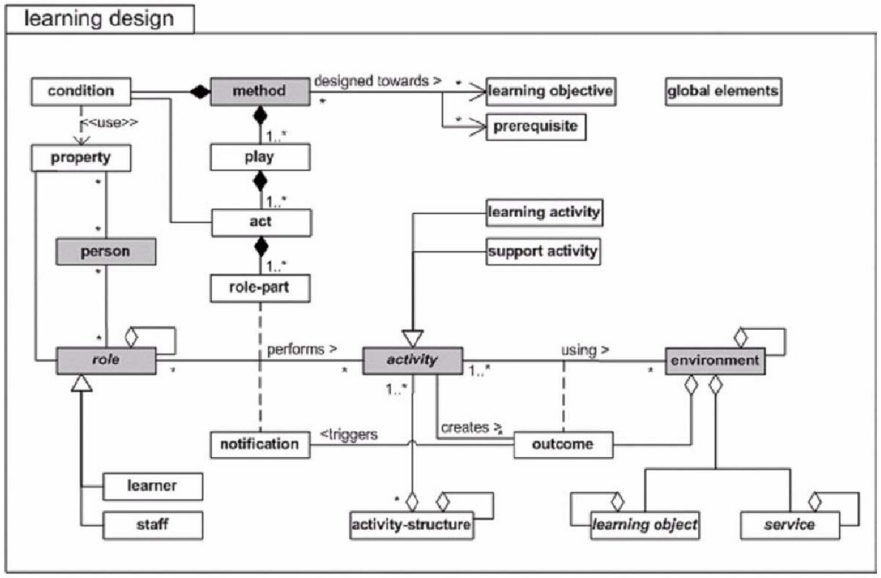

Fig. 1. IMS-LD MetaModel from [7]

As shown in Fig. 1, the core elements are activity, environment, role and method.

- the central point of IMS-LD is the *activity*. An activity defines the work to do within an *environment*. An activity is assigned to a *role*.
- The *environment* defines the learning objects which can be digital or not and the services provided like a chat or a forum.
- The *role* specifies the participants of a learning-design. The roles are divided in two groups, learner and staff. These roles can be sub-typed.
- The *method* determines the global objectives and the prerequisites to begin the learning-design. It also contains the *play* which defines the learning process, i.e., the sequence in which the activities are made and the corresponding roles. A play is defined as a theatrical play with acts.

3 Requirements for learning scenarios enactment

As explained in the above section, EMLs are becoming a common way to express learning scenarios because they focus on the pedagogical aspect, more than on the resource aspect. They also offer some great advantages like reusability and the use of high level concepts that can be understood and manipulated easily by a teacher. However, enacting these scenarios in a software platform is still an issue. What are the main requirements for this ? First, the system must be fully compliant with one or more EMLs but this is implicit. The two main problems are the possibility to redefine a part of the scenario during its execution and moreover to enable the users to participate in the changes. This is necessary because it is sometime difficult to fully express a scenario in details and some parts can change to adapt to the execution context. The context can evolve globally, changing laws for example, or locally, adapting to the need and the difficulties of a student. This is also in agreement with the work done in the field of CSCW and with Activity Theory [9] which promote a user-centered and tailorable platform where the user is able to redefine the model and the execution behavior The others requirements from our point of view are a support for cooperation, notably during the changing phases and support for standardised pedagogical ressources (like LOM and SCORM).

One category of software platform that can handle all these requirements can be flexible workflows systems. So in the next section, we will present theses systems.

4 Workflow systems

4.1 Presentation

To define a workflow, we use the following definition from [14]:

> *"Workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal."*

Usually, workflow systems are used in administrative domain where procedures are well defined and don't suffer exception or dynamic redefinition. This is mainly due to the fact that original workflows systems were heavily monolitic non flexible systems. Current trend in workflow research try to resolve this problem by working on flexible and dynamically redefinable system at run-time.

4.2 The workflow reference model

The workflow reference model is a standard from the Workflow Management Coalition (WfMC) whose aim is to promote the use of workflow through standardization and interoperability. This standard does not define the workflow engine itself but rather its interfaces as shown in figure 2. In a certain way, standardization in the e-learning domain follows the same principle, the LMS are not defined but all the elements used by an LMS are, like pedagogical ressources or EML.

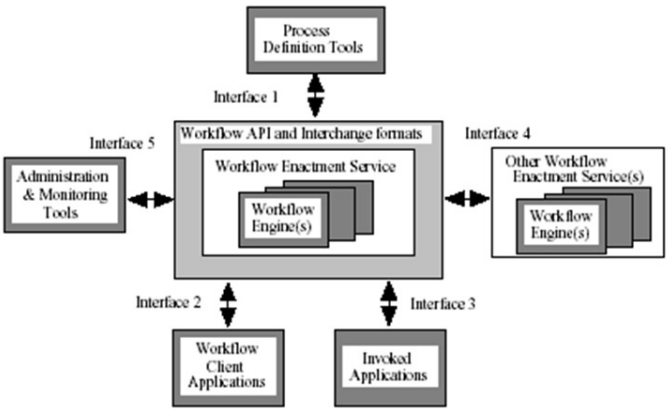

Fig. 2. WfMC reference model [14]

The workflow interfaces are the following:

Interface 1 defines an XML language called XPDL (*XML Process Definition Language*) [13] to define process models regardless of the enacting platform. It enables the use of different modelling tools as long as they can output XPDL;

Interface 2 is used by client applications to access the workflow services;

Interface 3 defines how the workflow engine can instantiate tools used in activities and get results from them;

Interface 4 defines an XML language for workflow interoperability. It enables one engine to create subprocesses in another engine and get the results;

Interface 5 is used by administration and monitoring tools to interact with the workflow.

In order to compare workflow modelling and IMS-LD, we will now present the XPDL.

4.3 XPDL

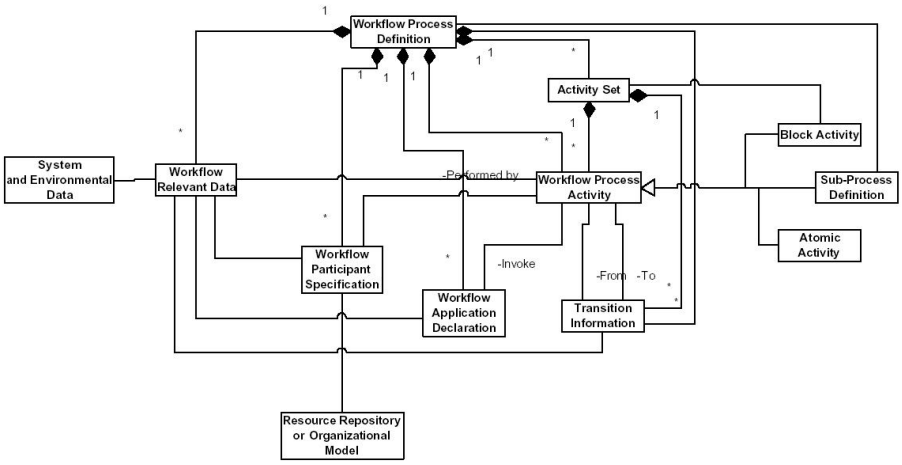

Fig. 3. XPDL MetaModel from [13]

As we have explained in the above section, IMS-LD expresses "learning workflow". Is this learning flow really different from work flow ? To answer this question, we will explain the meta-model of workflow systems (fig. 3), published by the Workflow Management Coalition (WfMC). The 5 top-level entities are Process, Activity, Transition, Relevant Data and Participant.

- *Process* defines the way to achieve a common goal, i.e., the path between the differents activities. It determine the execution context (overall description, input values, ...). This element is the container of all the other entities of the metamodel.
- *Activity* defines the work to realize; There are three types of activities. Subflow activity allows to execute another sub-process, block activity consist of an activity set which is an aggregation of activities and transitions and atomic activity is the real work to do. At the execution time, this work is transformed into work items which are executed by participants and/or applications.
- *Transitions* are the links between different activities. They define the control flow inside the process;

- *Relevant Data* are the data used and produced by the process and the activities. This data can be linked with the data of the enterprise, as for example an LMS system;
- *Participant* represents a human, role or group to whom work items are assigned. Participant can be linked with the organizational model of the enterprise.

After the presentation of EMLs and workflow systems, we will now present our platform for the enactment of learning scenarios.

5 The COW platform

COW (Cooperative Open Workflow) is a flexible workflow system developped in the Trigone laboratory which aims to enact learning path in LMS.

5.1 Requirements

This work has been driven by the following requirements:

- Support different learning styles. The workflow must be able to handle both individual and group work even within the scope of a single process;
- Support collaborative activities. We believe that collaborative learning can be a way to enhance the learning experience and strengthen learners' motivation;
- Support dynamic redefinition of the learning path. A tutor should be able to add activities for example if he detects a weakness in the learners knowledge. There are even some times when one does not even know beforehand the activities that will be needed;
- Support for reuse of existing course and activities models. It will be easier for pedagogical engineers to build course modules from existing parts. So we would like to provide predefined models (or skeletons) and also keep track of the models that have been modified during a course.

From a technical point of view, we would conceive a "learning scenario enacting component" which can be included within existing LMS. And for more interoperability, we would also following the existing standards for the implementation of the workflow engine. In the next sections, we describe the work done in the Cooperative Open Workflow to meet these requirements.

5.2 Workflow Architecture

The implementation respects the Workflow Management Facility specification [15] of the OMG and the workflow reference model of the WfMC [14].

Workflow Management Facility The Workflow Management Facility (WMF) [15] standardizes the architecture of the workflow engine. It has been defined by the Object Management Group (OMG) in accordance with the reference model of the WfMC. This standard defines entities such as WfProcess and WfActivity or WfResource which correspond to the elements described in a process model like XPDL. We have made an implementation of the interfaces proposed in th WMF with extensions to support collaborative activities and enhance the reuse of models.

an open micro-kernel architecture For the implementation, we have chosen a micro-kernel architecture. Our kernel offer the base fonctionnality of a workflow system. It was build using the MetaObject Protocol (MOP) [16]. This approach to obtain flexibility is also used in CSCW [17]. We can realize two types of modification in COW at run-time, the process model and the engine behavior.Process modification consist in adding/deleting/changing activities and transitions. These changes can be realized for one person or for a group of learner. Behavior modification consist in adapting the strategy to local context, like how react when an exception occurs ? The solutions can be stop the activity or send an e-mail, ... Thanks to this architecture, we can easily developp specific components to adapt the engine to specific needs. The integration within LMS systems is realised with webservices approach. Each interfaces of our sytem is accesible by using the SOAP [21] protocol.

User aspects Although we don't focus on user interaction with a a workflow engine, we have realised for demonstration purpose, a multi-channel access to our system. The engine outputs information in an XML format that we can transform to support multiple presentation such as HTML, WML or even VoiceXML using XSL stylesheets. A more detailled presentation can be found in [23].

6 Workflow systems in virtual campuses

There are few virtual campused that use a workflow engine to schedule learning activities. We review two of them hereafter.

6.1 Flex-el

The Flexible e-learning system (Flex-eL) [11] has been realized by the Distributed Systems Technology Center (DSTC) [12] and the university of Queensland in Australia. It consists of a distance education platform supported by a flexible workflow system. The focus of this work is on supporting the learning process rather than technology with the aim of designing a new learning environment to support new ways of learning. There a learning process for each student. This allow an easy way to adapt the learning path to the real need. Groups of students are dynamically constructed. The project started in march

2000 and the environment is currently tested at the University of Queensland in a course of a Master of Information Technology. According to the authors the first results seem positive since there are few abandons.

6.2 Campus Virtuel

The virtual campus platform named Campus Virtuel [18] developed by Archimed SA [19] also integrates a workflow system to handle the activities of a course module (figure 4).

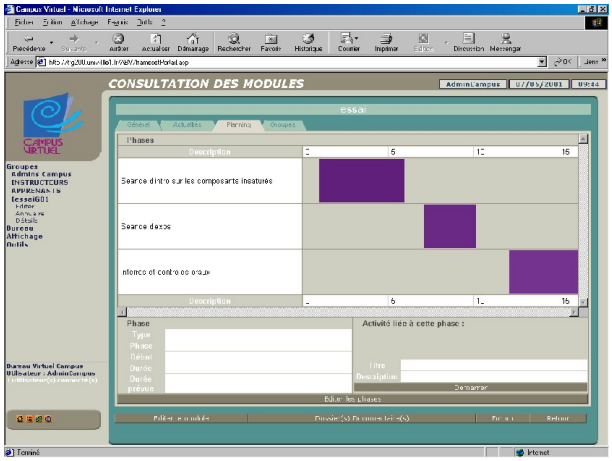

Fig. 4. Course model in the Campus Virtuel

A teacher or tutor can break his course into activities and associate the documents to use to each activity. The system is backed by an Electronic Documents Management System to manage and provide the pêdagogical ressources in the different activities. The system is limited by the fact that it cannot handle individual work since every student in a group must have terminated the current activity to be able to access the next one, even though some of the activities could be realized individually.

7 Linking Educational and Workflow Modellings

To enact a learning model in a workflow system, we have to translate IMS-LD into XPDL. As we have described in the above sections the metamodel of the two languages, we can now draw links between the entities. A simplified view of the links between IMS-LD and XPDL is given in Fig. 5.

The concept of activity is relatively identical. We have a direct link between an activity in IMS-LD and an activity in XPDL. Some attributes of

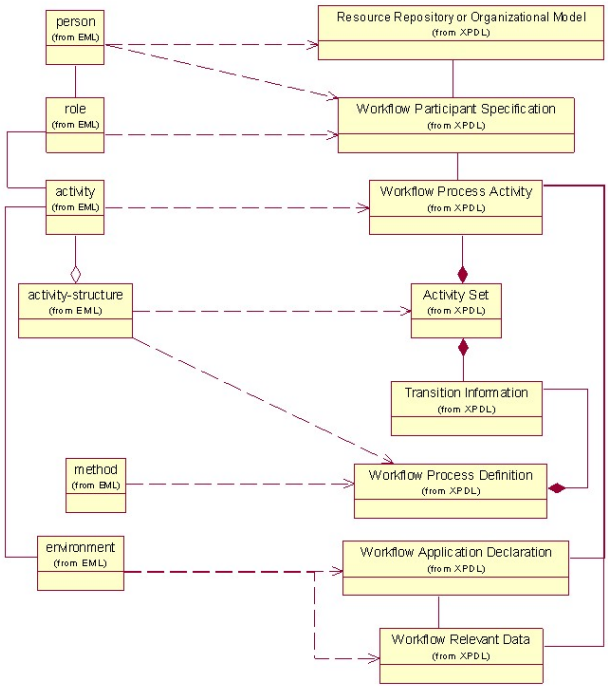

Fig. 5. Links between IMS-LD and XPDL

Activity$_{IMS-LD}$, like *learning-objectives* cannot be translated directly because these notions do not exist in a standard workflow system. To solve this problem, we used an element of the XPDL named *extended attribute*, designed to add specific vendor information in the XPDL. Hence, we avoid the loss of information during the transformation. And these information can be used by the LMS.

The concept of activity-structure is not exactly an activity-set because there isn't a self aggregation link for activity-set, i.e., an activity-structure can contains links to other activity-structure while the activity-set can not contain links to other activity-sets. In order not to lose this link, an activity-structure which contains link to another activity-structure is mapped to a workflow process.

The environment in IMS-LD can be links to workflow applications and to workflow data. The services are applications.

The role$_{IMS-LD}$ can be mapped to participant$_{XPDL}$ with a role type.

The informations of the method$_{IMS-LD}$, including play$_{IMS-LD}$, are partially mapped to process$_{XPDL}$ and transitions$_{XPDL}$.

To use IMS-LD models in our platform, we are implementing a transformer to pass from IMS-LD to XPDL which is the language used by COW. The rules

of transformations have been realized by using XSLT, because the two languages are based on XML.

Now, we will explain how models are enacted in COW.

8 Course models and instances

The main function of the workflow system is to schedule the activities of a pedagogical module. Such a module is attended by a group of students (ranging from 1 to n).

In the sequel, we will take a course in physics as an example of the modelling of a module. This module is broken into four activities described hereafter:

- *course learning* activity associated to the role *learner*;
- *exercises* activity associated to the role *learner*;
- *exercise correction* activity associated to the role *tutor*;
- *discussion about the module* activity associated to the role *learner* and *tutor*.

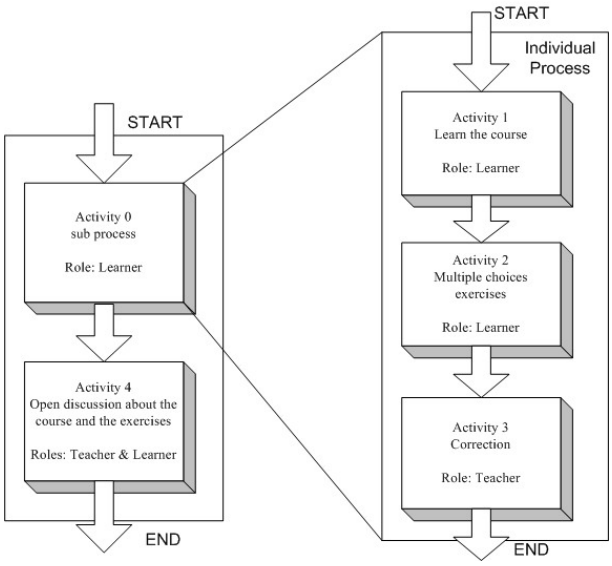

Fig. 6. Modelling of a pedagogical module

8.1 Course model

Since some parts of the module can be realized at his own rythm by each student, one has to take in into account so as to enable flexibility in the schedule of the

activities. There are two ways to manage the schedule of the activities for a group of students :

- In the first mode, an activity is terminated only when all the students have terminated it. In such a way, the activities of a whole group are synchronised. Even though it is very close to traditional face to face learning, it does not take benefit of distance learning mode;
- The second mode identifies the parts of a module that can be realized autonomously. This way each student can progress at his own rythm inside a group with some activities giving a synchronization point to the group.

These two modes are supported in COW by the mean of sub-processes. In our scenario, the teacher decides that the three first activities can be realised individually by each student. These activities are then modelised into a single process. The process corresponding to the whole module is then composed of two sequential activities (see figure 6). The first one being in fact a reference to the individual work sub-process and the second one corresponding to a synchronous discussion beetwen the members of the group.

Collaborative activities To handle collaborative activities, we have made some modifications to XPDL and the WMF to introduce the notion of *workitem*. A workitem is an atomic piece of work and an activity is composed of workitems. In the simplest case, there is only one workitem in an activity. However, within a collaborative activity, there can be more workitems. A workitem is attributed to a role, so if multiple actors have the same role, there will be an instance of the workitem for each of them in the activity. Resources are allocated to the workitems rather than the activity. In our example the discussion activity could be done with a chat tool but the learners and the tutor may not have the same rights on the tool since they do not have the same roles.

Time constraints Management of time constraints is an important aspect of learning activities. This is particularly true for group based learning where there must not be too much lag between the learners. COW supports the notion if *deadlines* which correspond to the time at wich an activity must be started or completed. It also support the notion of *limit* which defines the minimal and maximal duration of an activity. When a stop deadline of a maximum limit is reached, the workflow engine suspends the activity. It can then use different policies to handle the case. For example, the system can terminate the activity authoritavely or notify the tutor who will take a decision about it. These behaviors can be dynamically change at run-time to be adapted to a changing context by the tutor.

8.2 Course instance

The creation of a process instance requires an instance model which describes the mapping of roles to actual users and of resources to tools. The mappings can

be global to the model or defined on an activity basis. This separation between process model and instance data allows for a better reuse of models. Figure 7 illustrates the workflow operation.

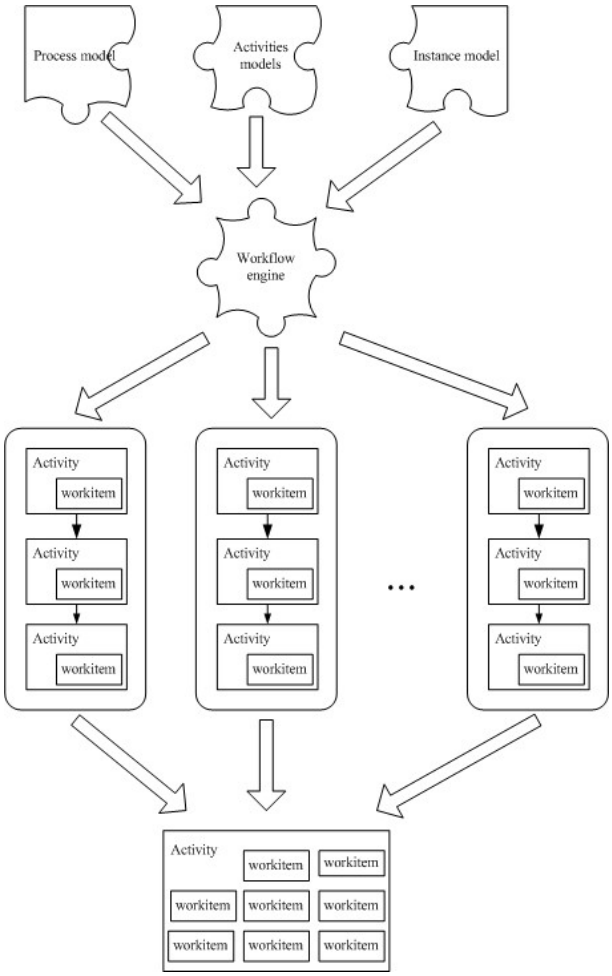

Fig. 7. course instantiation

Taking the process and activities models and isntance data, the workflow engine will create a subprocess for each learner. These subprocesses contains the three activities that can be performed individiually and each activity contains only one work item. The third activity is performed by the tutor role will have three workitems to do from different processes. When all subprocesses are ter-

minated, the engine will create a collaborative activity with one workitem for each learner and one for the tutor.

9 Conclusion and outlooks

In this paper, we have presented how it can be possible to execute IMS-LD models by using a workflow approach. We have compared the two metamodels and realized the mapping between them. Thanks to this comparison, we fave realized a model transformer to translate IMS-LD into XPDL, that will allow us to enact IMS-LD models with COW, our flexible workflows engine designed for the educational field. From our point of view, this approach is interesting because we use the concepts of two different worlds. We think workflow technology is a great tool for the management of learning activities inside a virtual campus. Our engine is currently being integrated within the Campus Virtuel from the Archimed Company for their next relesase which will enable us to start realistic experimentations.

There is still some unresolved problems and questions. How the prerequisite can be handled by the workflow system as a constrainst between different process models and knowing these prerequisites and the wishes of a student, how can we automaticaly construct a learning path ? When we dynamically change the process model, our system saves changes in XPDL. In order to show the modification to a pedagogical engineer we must export XPDL into IMS-LD, but we haven't yet realize this transformation. Our future work will try to answer these questions and considering the use of an MDA approach to generate the implementation of the learning paths from the IMS-LD models.

Acknowledgements

Research on workflow system has been funded by Archimed SA (`http://www.archimed.fr`).

References

1. OMG Model Driven Architecture, `http://www.omg.org/mda`.
2. Object Management Group. `http://www.omg.org`.
3. Miguel Rodriguez-Artacho. "PALO Language Overview", technical report LSI Dept/TR-2002-01, Universidad Nacional de Educatión a Distancia, january 2002. `http://sensei.lsi.uned.es/palo/PALO-TR.pdf`.
4. Educational Modelling Language. Open University of Nederland. `http://eml.ou.nl`
5. Dan Brickley. "Towards an open question-interchange framework", University of Bristol, `http://www.ilrt.bris.ac.uk/netquest/about/lang/motivation.html`.
6. CEN/ISSS WS/LT Learning Technologies Workshop. "Survey of Educational Modelling Languages (EMLs)", version 1, 19 september 2002.
7. IMS Global Learning Consortium. "IMS Learning Design Information Model", version 1.0 Final Specification, 20 january 2003. `http://www.imsproject.org/learningdesign/index.cfm`.

8. IMS Global Learning Consortium. "IMS Question & Test Interoperability: An Overview", version 1.2, 11 february 2002, http://www.imsproject.org/question/index.cfm.
9. Gégrory Bourguin, Alain Derycke and Jean-Claude Tarby. "Beyond the Interface: Co-evolution Inside Interactive Systems - A Proposal Founded on Activity Theory", in proceeding of IHM-HCI 2001, Lille, France, September 2001.
10. Thomas Vantroys and Yvan Peter. "Un système de workflows flexible pour la formation ouverte et à distance", in proceedings of TICE2002, november 2002. http://tice2002.insa-lyon.fr.
11. Joe Lin, Charley Ho, Wasim Sadiq, Maria E. Orlowska, "On Workflow Enabled e-Learning Services", *In Proceedings of the IEEE International Conference on Advanced Learning Technologies*, ICALT2001, 6 - 8 August 2001, Madison, USA.
12. Distributed Systems Technology Centre http://www.dstc.edu.au.
13. Workflow Management Coalition. "Workflow Process Definition Interface – XML Process Definition Language", WfMC-TC-1025, version 1.0, 25 october 2002. http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf
14. Workflow Management Coalition. "The Workflow Reference Model", WfMC-TC-1003, Version 1.1, 19 january 1995. http://www.wfmc.org/
15. Object Management group. "Workflow Management Facility",Version 1.2, 2000. http://www.omg.org/technology/documents/formal/workflow_management.htm.
16. Gregor Kiczales, Jim des Rivières et Daniel G. Bobrow – The Art of the Metaobject Protocol. MIT Press, septembre 1991. ISBN : 0-26-261074-4.
17. Gégory Bourguin. "Un support informatique à l'activité coopérative fondé sur la Théorie de l'activité : le projet DARE", Ph'D Thesis, Université des Sciences et Technologies de Lille, 13 july 2000, Lille, France.
18. Claude Viéville. "Learning Activities in a Virtual Campus", chapter 15 in *The Digital University - Building a learning Community*, Reza Hazemi and Stephen Hailes (Eds), Springer, pages 215-227, 2002.
19. Archimed SA. http://www.Archimed.fr.
20. Linda G. DeMichiel, L. mit Yalçinalp and Sanjeev Krishnan. "Enterprise JavaBeans^{TM} Specification", version 2.0, Sun Microsystems, Inc., 2001.
21. W3C. XML Protocol specification. http://www.w3.org/2000/xp/.
22. W3C. Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl.
23. Thomas Vantroys and José Rouillard – Workflow and Mobile Devices in Open Distance Learning. In proceedings of IEEE International Conference on Advanced Learning Technologies ICALT 2002, Kazan, Tatarstan, 9-12 septembre 2002. http://lttf.ieee.org/icalt2002/.